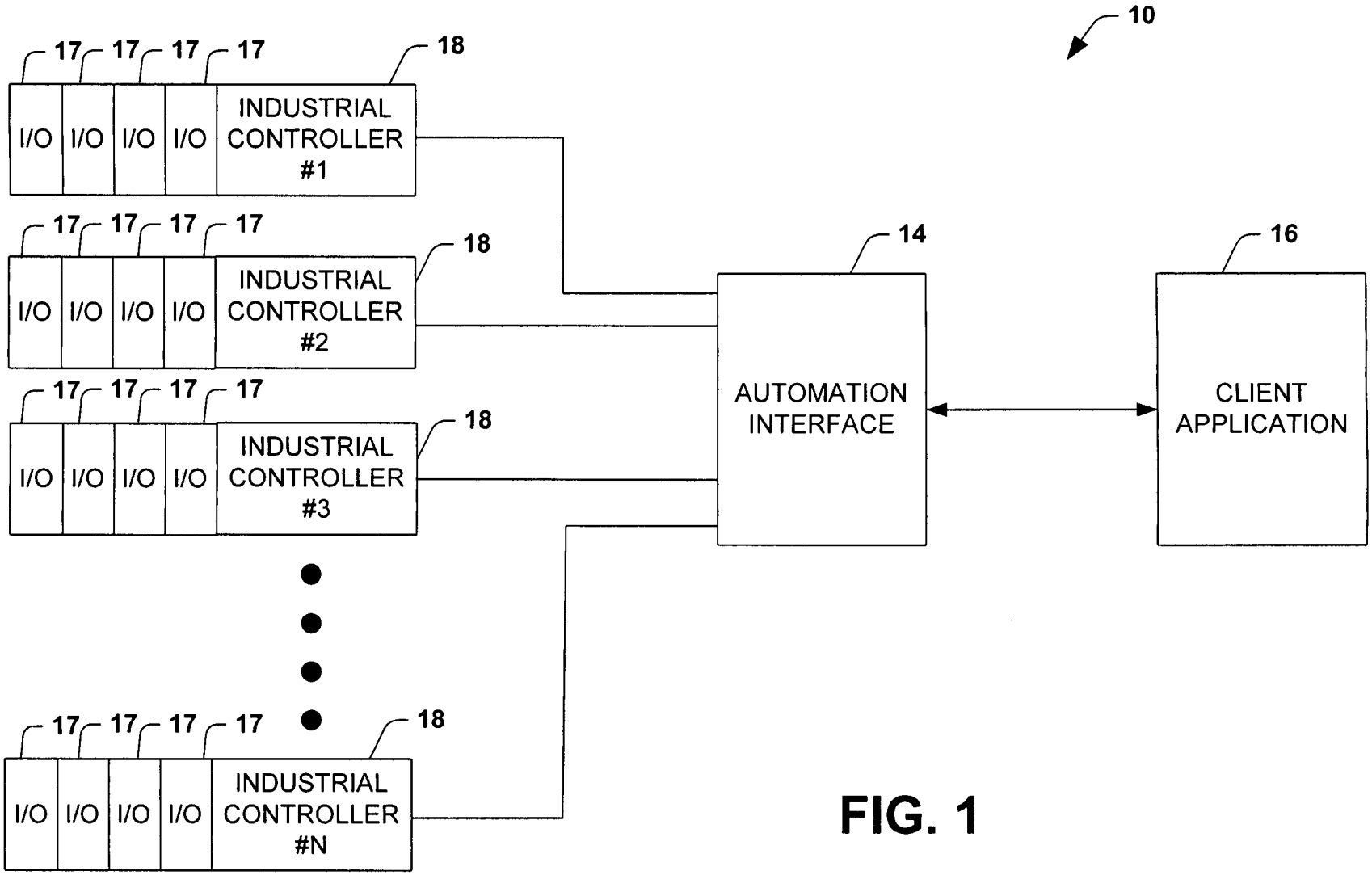

10
17
17
17
17
18
I/O
I/O
I/O
I/O
INDUSTRIAL CONTROLLER #1
INDUSTRIAL CONTROLLER #2
INDUSTRIAL CONTROLLER #3
INDUSTRIAL CONTROLLER #N
14
AUTOMATION INTERFACE
16
CLIENT APPLICATION

FIG. 1

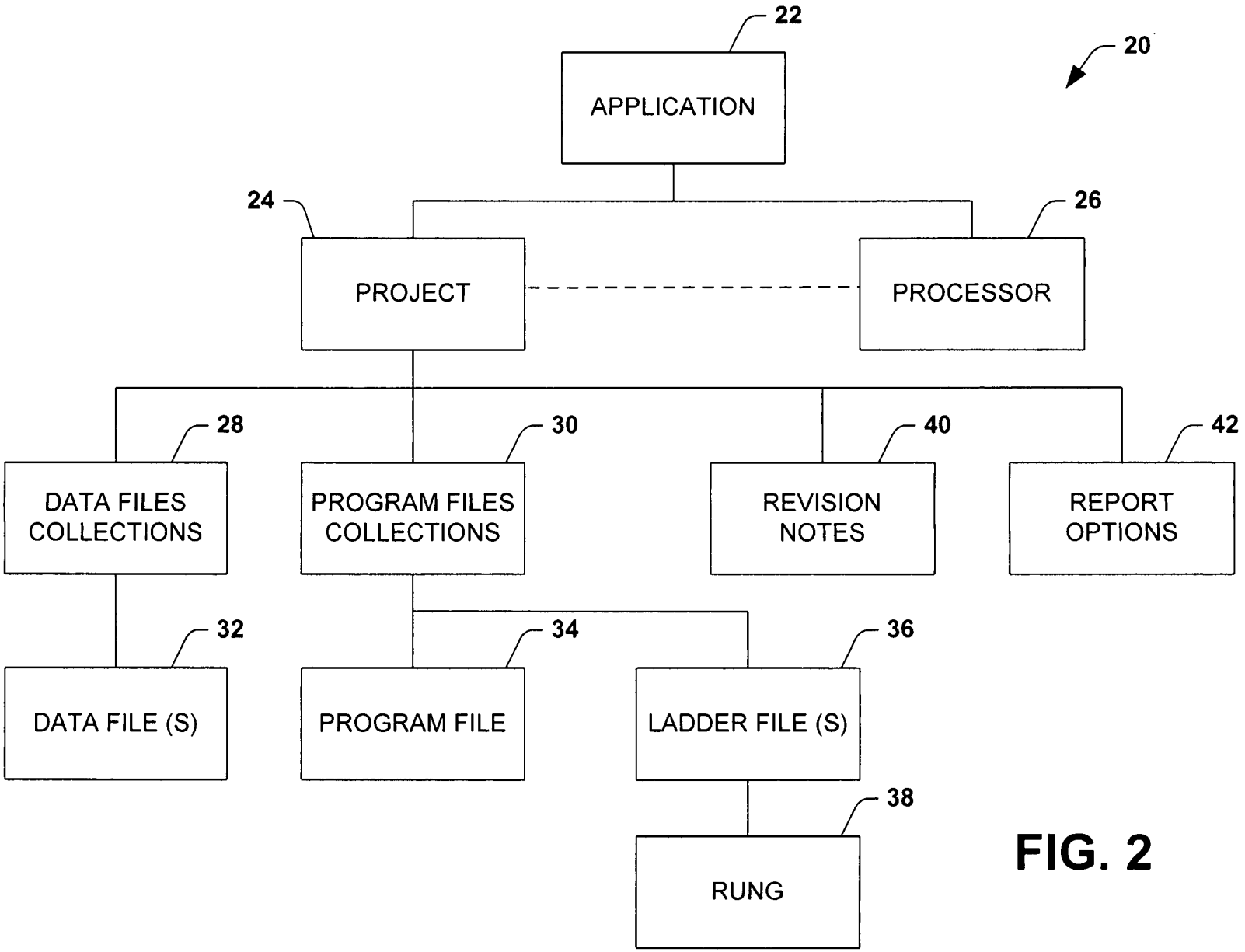

22
20
APPLICATION
24
26
PROJECT
PROCESSOR
28
30
40
42
DATA FILES COLLECTIONS
PROGRAM FILES COLLECTIONS
REVISION NOTES
REPORT OPTIONS
32
34
36
DATA FILE (S)
PROGRAM FILE
LADDER FILE (S)
38
RUNG

FIG. 2

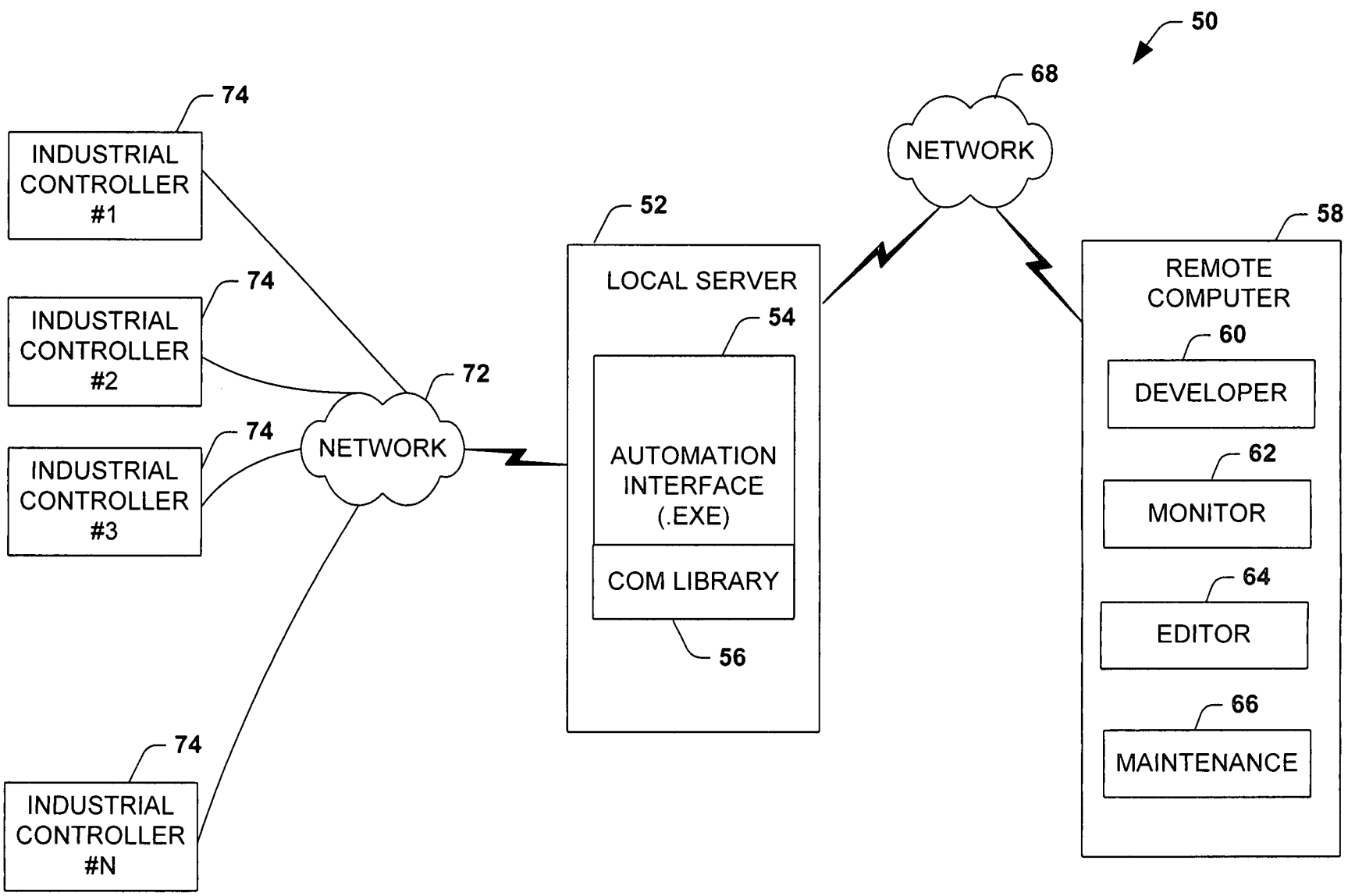

50
68
NETWORK
74
INDUSTRIAL CONTROLLER #1
74
INDUSTRIAL CONTROLLER #2
74
INDUSTRIAL CONTROLLER #3
74
INDUSTRIAL CONTROLLER #N
72
NETWORK
52
LOCAL SERVER
54
AUTOMATION INTERFACE (.EXE)
COM LIBRARY
56
58
REMOTE COMPUTER
60
DEVELOPER
62
MONITOR
64
EDITOR
66
MAINTENANCE

FIG. 3

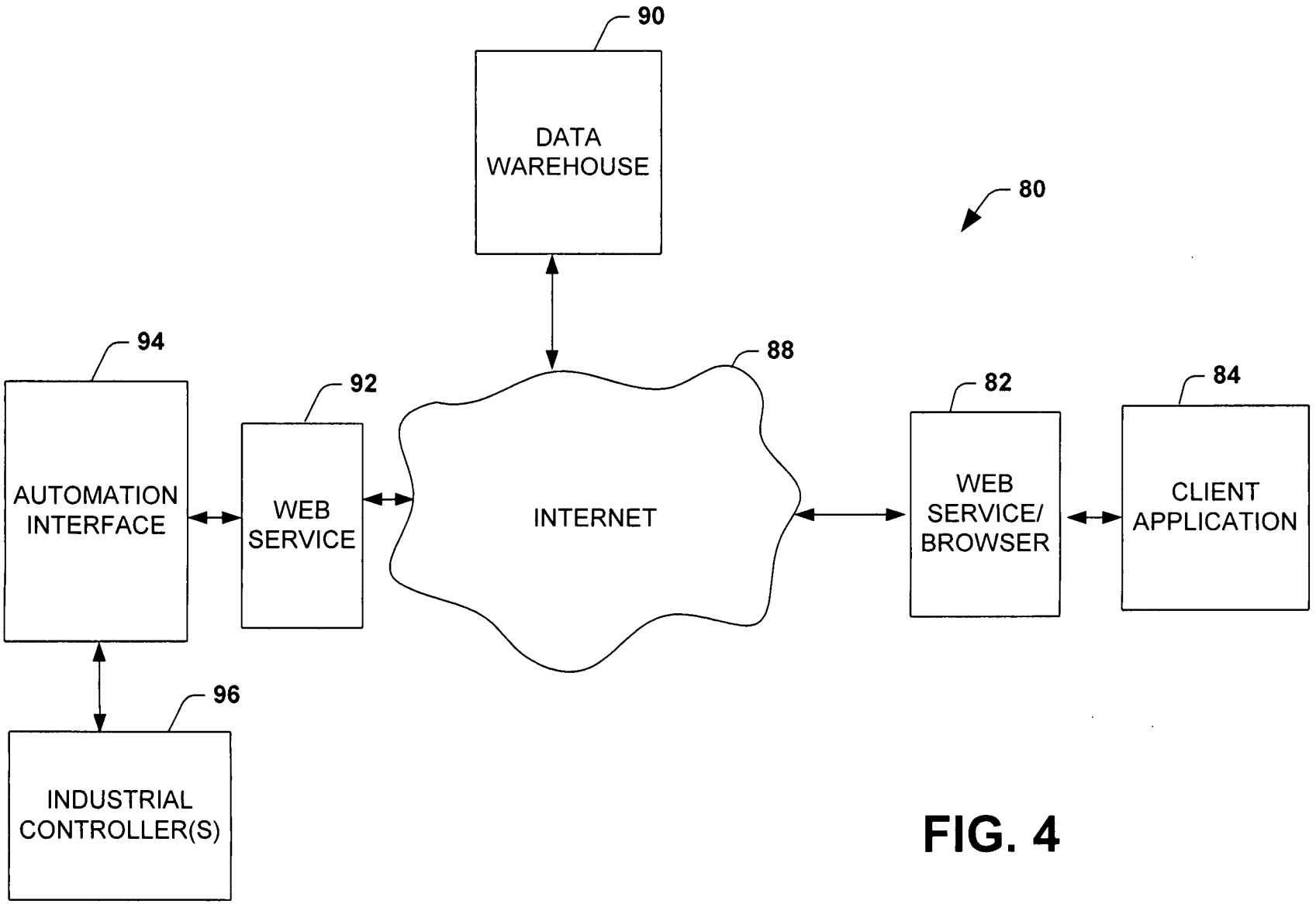
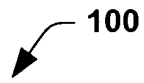
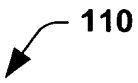

90
DATA WAREHOUSE
80
94
92
88
82
84
AUTOMATION INTERFACE
WEB SERVICE
INTERNET
WEB SERVICE/ BROWSER
CLIENT APPLICATION
96
INDUSTRIAL CONTROLLER(S)

FIG. 4



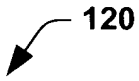

```
01  Public Sub DownloadAProcessor()
02      Dim g_Application As AutoInterface.Application
03      Dim g_Project As AutoInterface.Project
04      Dim EncodedRouteData As String
05
06      '** start Project and store it in the g_Application object variable
07      Set g_Application = CreateObject("Project.Application")
08
09      '** Declare some variables for clarity.
10      Dim ShowDialog As Boolean
11      Dim UseAutoSave As Boolean
12      Dim IgnorePrompts As Boolean
13      Dim OnlineAction As IgxOnlineAction
14
15      '** Initialize these variables to suitable defaults
16      ShowDialog = False
17      UseAutoSave = False
18      IgnorePrompts = True
19      OnlineAction = IgxGoOffline
20
21      '** Open a project from disk that will be downloaded.
22      Set g_Project  = g_Application.FileOpen( "C:\Projects\Upload.rsp", ShowDialog, UseAutoSave)
23
24      '** The communication route to the actual processor that this project represents is set up in the
25      '** application when the project is opened. Get this info out now and save it away for later calls
26      '** to upload this processor.
27      EncodedRouteData = g_Project.EncodedRouteString
28      SaveEncodedRouteData EncodedRouteData
29
30      '** Now download the image to the processor.
31      Dim NoError As Boolean
32      NoError  = g_Project.Download(IgnorePrompts, OnlineAction, IgxREMOTEPROG)
33  End Sub
```

FIG. 5




```
01 Public Sub UploadAProcessor()
02      Dim g_Application As AutoInterface.Application
03      Dim g_Project As AutoInterface.Project
04      Dim EncodedRouteData As String
06
07
08      '** start Project and store it in the g_Application object variable
09      Set g_Application = CreateObject("Project.Application")
10
11      '** EncodedRouteData holds a string of data that points the project software to the processor on
12      '** the Data Highway+ network it will be performing uploads and downloads on. A function is
13      '**called here to initialize it to a previously obtained value
14
15      InitializeRouteData EncodedRouteData
16
17      '** Now tell project which processor it is to communicate with
18      g_Application.EncodedRouteString = EncodedRouteData
19
20      '** Declare some variables for clarity.
21      Dim IgnorePrompts As Boolean
22      Dim SaveChanges As Boolean
23      Dim AcceptDefaultAction As Booloen
24      Dim UploadAction As lgxUploadDownloadAction
25      Dim OnlineAction As lgxOnlineAction
26      Dim DataBaseAction As lgxSaveAction
27
28      '** Initialize these variables to suitable defaults
29      IgnorePrompts = True
30      SaveChanges = False
31      AcceptDefaultAction = True
32      UploadAction = lgxUploadCurrent
33      OnlineAction = lgxGoOffline
34      DataBaseAction = lgxNoAction
35
36      '** Now make the call that performs the upload. Store the uploaded project object
37      Set g_Project = g_Application.Upload(IgnorePrompts,  SaveChanges, UploadAction,
38              _OnlineAction)
39
40      '** Now save the uploaded image to disk as a file called Upload.rsp
41      Dim NoError As Boolean
42      NoError  = g_Project.SaveAs(IgnorePrompts, AcceptDefaultAction, _
43              DataBaseAction, "C:\Projects\Upload.rsp")
44 End Sub
```

FIG. 6




```
01      Public Sub InsertLadderRung()
02      Dim g_Application As AutoInterface.Application
03      Dim g_Project As AutoInterface.LogixProject
04      Dim g_ProgFile As AutoInterface.ProgramFile
05      Dim g_LadderFile As AutoInterface.LadderFile
06
07      '** start AutoInterface and store it in the g_Application object variable
08      Set g_Application = CreateObject("AutoInterface.Application")
09
10      '** Declare some variables for clarity.
11      Dim ShowDialog As Boolean
12      Dim UseAutoSave As Boolean
13      Dim IgnorePrompts As Boolean
14      Dim AcceptDefaultAction As Boolean
15      Dim OnlineAction As lgxOnlineAction
16
17      '** Initialize these variables to suitable defaults
18      ShowDialog = False
19      UseAutoSave = False
20      IgnorePrompts = True
21      AcceptDefaultAction = True
22      OnlineAction = lgxGoOffline
23
24      '** Open a project from disk that will be modified.
25      Set g_Project  = g_Application.FileOpen( "C:\Projects\Upload.rsp", ShowDialog, UseAutoSave)
26
27      '** Obtain the program file object for program file 2. Cast this object to a LadderFile object since
28      '** in this case, program file 2 is also a ladder file.
29      Set g_ProgFile = g_Project.ProgramFiles(2)
30      Set g_LadderFile = g_ProgFile
31
32      '** Build up a sample rung manually
33      Dim RungString As String
34      RungString = "SOR XIC B3:0/0 OTE B3:0/1 EOR"
35
36      '** Insert this rung into the ladder at position 0
37      g_LadderFile.InsertRungAsAscii 0, RungString
38
39      '** Save the modified project to disk first, then
40      '** download the image to the processor and set the processor to RUN mode.
41      '** Once this is completed, the new rung is executing in the processor.
42      Dim NoError As Boolean
43      NoError  = g_Project.Save(IgnorePrompts, AcceptDefaultAction)
44      NoError  = g_Project.Download(IgnorePrompts, OnlineAction, lgxREMOTERUN)
45  End Sub
```

FIG. 7

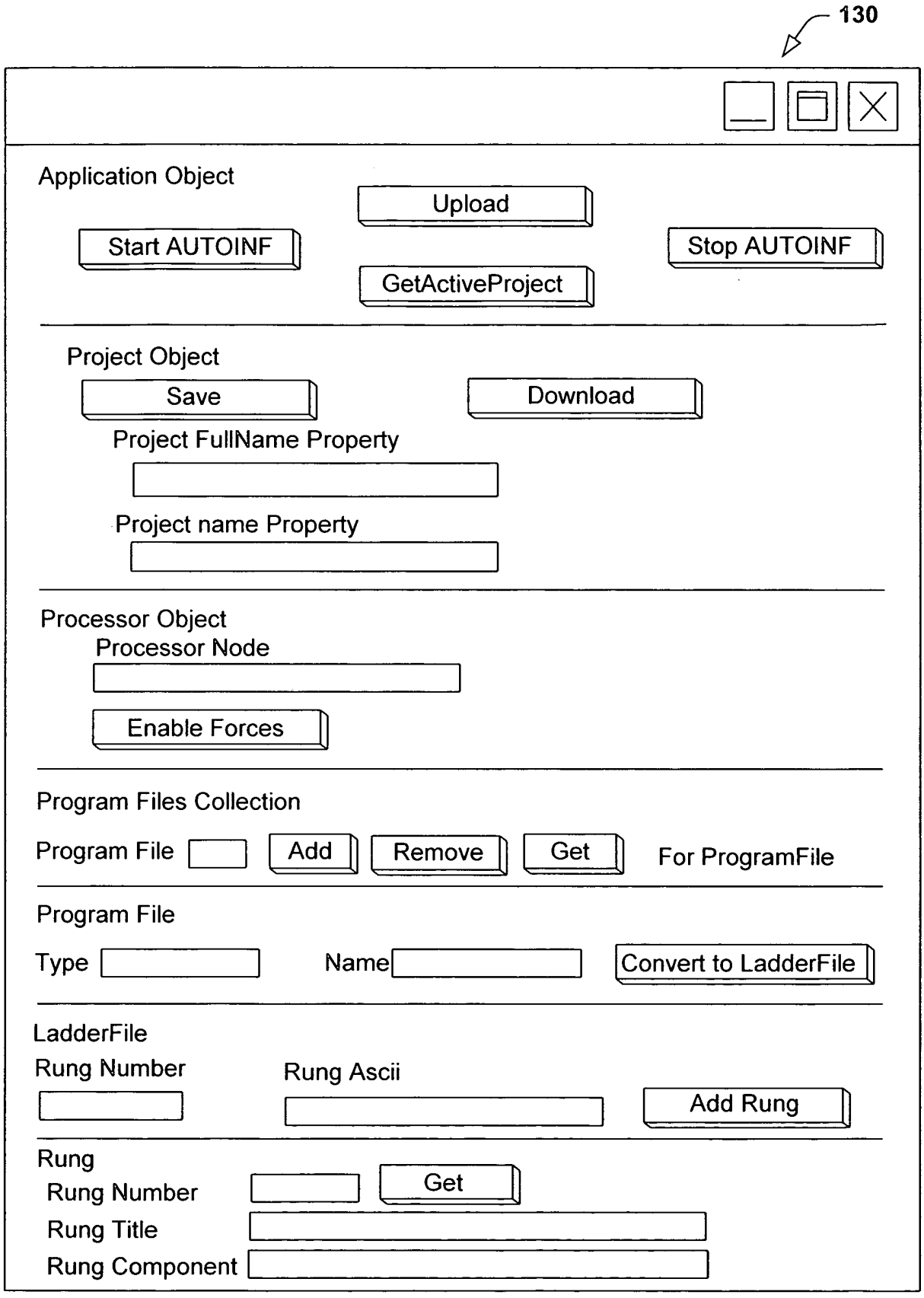

130
Application Object
Upload
Start AUTOINF
Stop AUTOINF
GetActiveProject
Project Object
Save
Download
Project FullName Property
Project name Property
Processor Object
Processor Node
Enable Forces
Program Files Collection
Program File
Add
Remove
Get
For ProgramFile
Program File
Type
Name
Convert to LadderFile
LadderFile
Rung Number
Rung Ascii
Add Rung
Rung
Rung Number
Get
Rung Title
Rung Component

FIG. 8

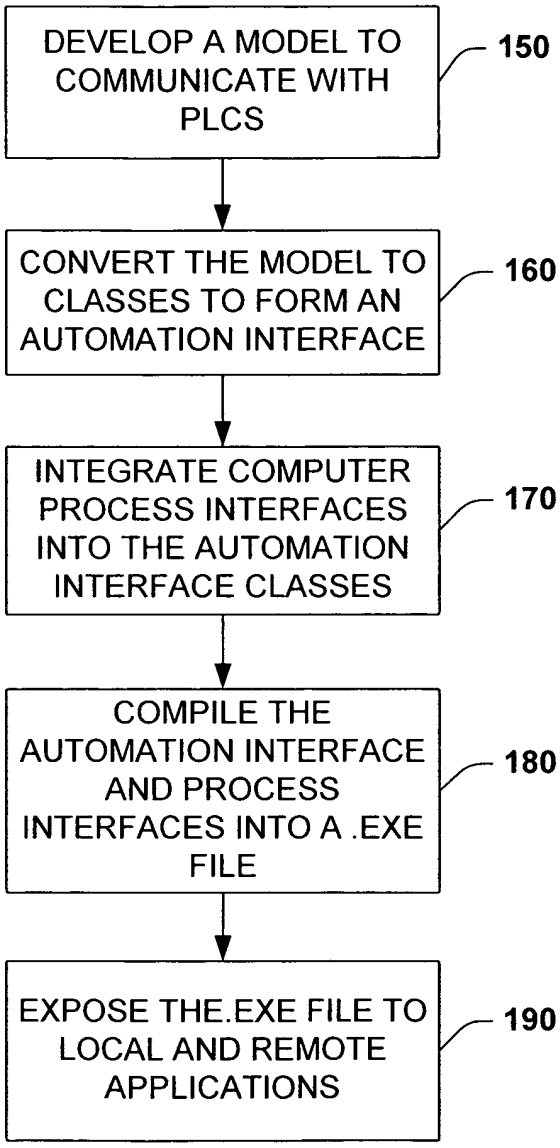

DEVELOP A MODEL TO COMMUNICATE WITH PLCS
150
CONVERT THE MODEL TO CLASSES TO FORM AN AUTOMATION INTERFACE
160
INTEGRATE COMPUTER PROCESS INTERFACES INTO THE AUTOMATION INTERFACE CLASSES
170
COMPILE THE AUTOMATION INTERFACE AND PROCESS INTERFACES INTO A .EXE FILE
180
EXPOSE THE.EXE FILE TO LOCAL AND REMOTE APPLICATIONS
190

FIG. 9

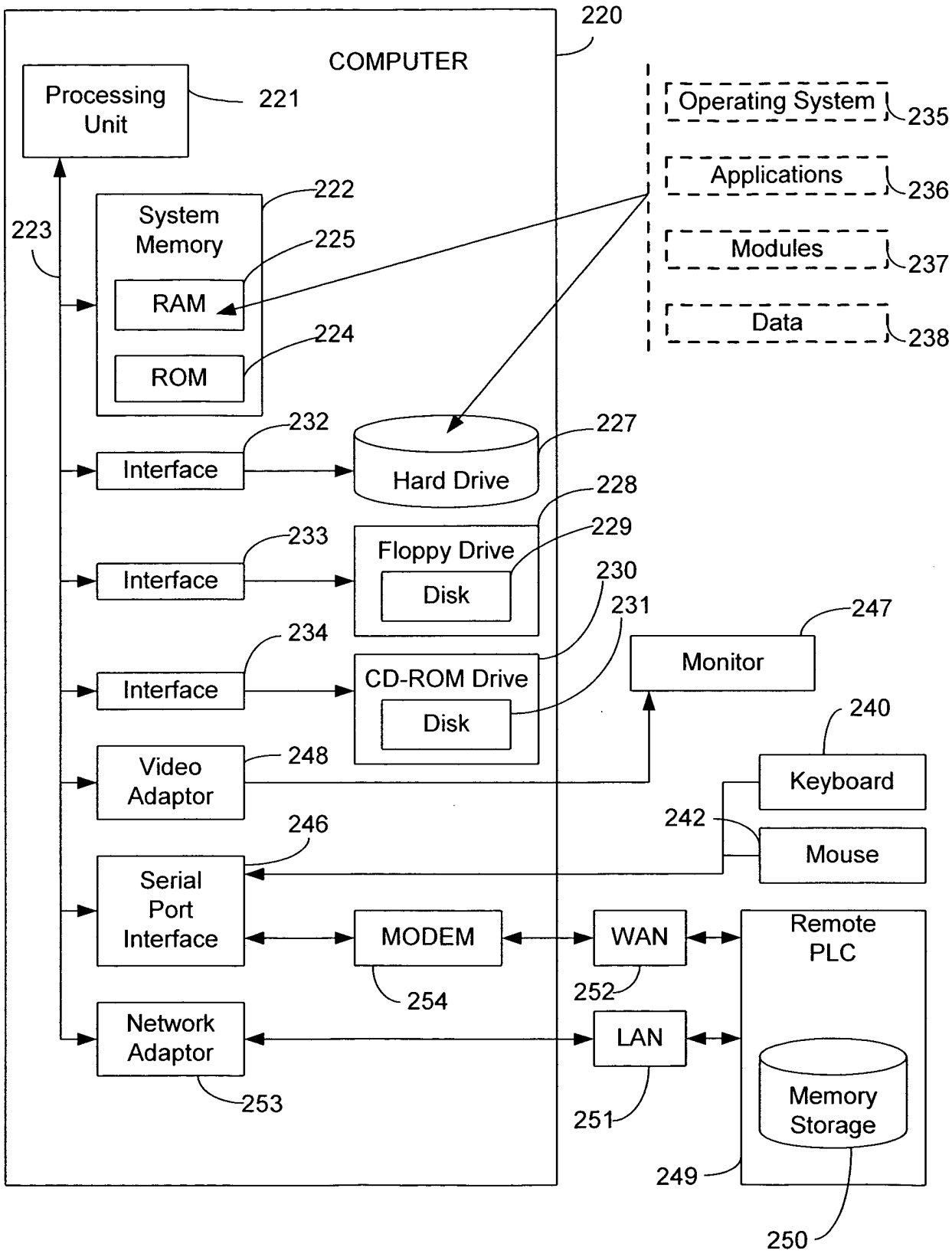

COMPUTER
220
Processing Unit
221
Operating System
235
Applications
236
Modules
237
Data
238
System Memory
222
223
225
RAM
224
ROM
232
Interface
Hard Drive
227
228
233
Interface
Floppy Drive
229
Disk
230
234
Interface
CD-ROM Drive
231
Disk
247
Monitor
240
Keyboard
242
Mouse
Video Adaptor
248
246
Serial Port Interface
MODEM
254
WAN
252
Remote PLC
Network Adaptor
253
LAN
251
Memory Storage
249
250

FIG. 10